Supplementary Material for “Incremental Development of Safety Cases: a Mapping Study”

CAMILO ALMENDRA*, Universidade Federal do Ceará (Campus Quixadá/UFC) and Universidade Federal de Pernambuco (CIn/UFPE)
CARLA SILVA*, Universidade Federal de Pernambuco (CIn/UFPE)
JÉSSYKA VILELA*, Universidade Federal de Pernambuco (CIn/UFPE)

This is a supplementary material for “Incremental Development of Safety Cases: a Mapping Study” paper. It presents more information on the search strategy, a quality evaluation of the studies, extended data for research questions RQ1 and RQ3, and additional research questions RQ5 and RQ6.

Additional Key Words and Phrases: Supplementary material

ACM Reference Format:
Camilo Almendra, Carla Silva, and Jéssyka Vilela. 2020. Supplementary Material for “Incremental Development of Safety Cases: a Mapping Study”. In ,. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACRONYMS

SAC Safety Assurance Case.
SACD Safety Assurance Case Development.
SCS Safety-Critical System.
SDLC Software Development Life Cycle.
SMS Systematic Mapping Study.

1 RESEARCH PROTOCOL

1.1 Search Strategy

The first part of the search string includes SAC-related terms to constrain the domain, and the second part includes development-related terms. In the string refinement process, we found that the term “safety-critical development” is not the only way used to refer to SCS development; for example, the term “system development” is also used. The construction and refinement of the search string were carried by the authors, all of them with experience in Software Engineering (SE) research. The third author, in particular, has experience in researching SE for SCS and in performing SLRs in this field. Moreover, the second author has been a researcher in SE for at least 18 years.

1.2 Threats to validity

Table 1 describes how integration between SACD and SDLC was referred in all the selected studies.

*All authors contributed equally to this research.

SBES ’20, SBES 2020, Natal, Brazil
© 2020 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Table 1. Terms used to denote development of SACs alongside the SDLC.

Study	Title or/and abstract excerpts
[14]	"common evolutionary process"
[6]	"Assurance Based Development"
	"synergistic construction"
	"Co-developing the system and its assurance case"
[4]	"Evidence-Based Development"
	"approach to progressive system assurance (...) integrated with the development process"
[7]	"Assurance Driven Design"
	"how software and assurance argument can be built together"
[1]	"evidence sources that can be used at the various stages of the lifecycle"
[5]	"develop both a software system and a safety argument iteratively"
[12]	"promote dependability through the lifecycle"
[8]	"safety case becomes the focus of safety engineering throughout the system lifecycle"
[10]	"how an assurance case can be generated when a software process is employed"
[15]	"Safety Case Driven Development"
[9]	"Continuously Revised Assurance Cases"
	"developing assurance cases (...) at every stage of the system life cycle"
[16]	"certification process to be done in parallel with development"
[2]	"Evidence-oriented V-model Methodology"
	"seamless and unified development process"
[13]	"assurance drives the engineering process"
[3]	"incrementally applied safety-related activities"
[11]	"incorporating agile practices into critical software development"

We list below some of the main safety engineering venues indexed by database engines used:

- International Conference on Computer Safety, Reliability and Security (SAFECOMP)
- European Conference on Dependable Computing (EDCC)
- International Symposium on High Assurance Systems Engineering (HASE)
- IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)
- International Symposium on Software Reliability Engineering (ISSRE)
- IEEE Transactions on Dependable and Secure Computing
- Reliability Engineering and System Safety Journal
- Journal of Reliable Intelligent Environments
- International Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems (SASSUR)
- International Workshop on Assurance Cases for Software-intensive Systems (ASSURE)

2 QUALITY ASSESSMENT

The quality assessment (QA) of selected studies in our SMS was achieved by a scoring technique to evaluate the credibility, completeness and relevance of the selected studies. All papers were evaluated against a set of seven quality criteria:

Q1. Is there a clear statement of the goals of the research [17]?
Q2. Is the proposed technique clearly described [17]?
Q3. Is there a discussion about the results of the study [17]?
Q4. Are the limitations of this study explicitly discussed [17]?
Q5. Is the article relevant for practitioners [22]?

Q6. Is there sufficient discussion of related works [22]?
Q7. Is the study significantly increase the knowledge about integration of SACD and SDLC research [22]?

The assessment instrument used is presented in Table **??**. Each quality assessment question is judged against three possible answers: “Yes” (score = 1), “Partially” (score = 0.5) or “No” (score = 0). Then, the quality score is computed by taking the sum of the scores of each answer.

Table 2. Quality scores and number of citations for each selected study.

Study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total	Qual.	Citations
Papadopoulos and McDermid [14]	1	1	0.5	0	1	0	0.5	4	57.1%	59
Graydon et al. [6]	1	1	0	0	1	0.5	0.5	4	57.1%	74
Dick and Wills [4]	1	0.5	0	0	1	0	0	2.5	35.7%	3
Hall and Rapanotti [7]	1	1	1	0	0	1	0.5	4.5	64.2%	36
Alexander et al. [1]	1	0.5	0	0	1	0	1	3.5	50%	13
Ge et al. [5]	1	1	1	1	1	0	1	6	85.7%	66
Matsuno et al. [12]	0.5	0.5	0	0.5	0.5	0.5	0.5	3	42.9%	21
Knight et al. [8]	1	1	0	0	1	0	1	4	57.1%	1
Lin and Shen [10]	1	1	1	0	1	1	1	6	85.7%	6
Ruiz et al. [15]	1	1	0.5	0	1	0.5	0.5	4.5	64.2%	2
Kuramitsu [9]	1	1	1	0	1	1	1	6	85.7%	0
Stålhane and Myklebust [16]	1	1	1	0.5	1	1	1	6.5	92.8%	11
Cicotti [2]	1	1	1	0.5	1	1	0.5	6	85.7%	2
O'Halloran et al. [13]	1	1	1	1	1	0	1	6	85.7%	4
Cleland-Huang and Vierhauser [3]	1	1	1	1	1	1	1	7	100%	7
Łukasiewicz and Górski [11]	1	1	0	0	1	0.5	0.5	4	57.1%	0
Average	0.97	0.91	0.56	0.28	0.91	0.50	0.72	4.84	69.2%	19.0

3 RQ1 – WHAT ARE THE CHARACTERISTICS OF THE APPROACHES THAT PROMOTE INCREMENTAL SACD INTEGRATED WITH SDLC?

3.1 Venus of publication

The 16 selected studies are from 14 different venues (conferences, journals, and book chapter) (see Table 3). Eight venues are related to reliability, safety and dependability research (V1–8), and six venues are related to software engineering and general computer science (V9–14). As the subject of this mapping covers two fields (safety engineering and software engineering), it was expected to find studies in venues of both scope. Only two venues had two studies selected: the Journal of Reliability Engineering and System Safety (V1) and the International Workshop on Assurance Cases for Software-Intensive Systems – ASSURE (V2). V1 is devoted to methods for the enhancement of the safety and reliability of complex technological systems, and V2 is mainly devoted to discussing the application of assurance case technology for supporting assurance of software-intensive systems.

4 RQ3 – WHICH ARE THE ROLES INVOLVED IN THE APPROACHES?

We found 25 roles cited in 12 studies (four studies did not explicitly discuss roles [4, 11, 14, 15]). Table 4 presents all the roles identified in the selected studies.

Table 3. Venues of publication.

Venue	Studies
V1: Reliability Engineering and System Safety (*journal*)	[14][13]
V2: Intl. *Workshop* on Assurance Cases for Software-Intensive Systems	[8][16]
V3: Intl. *Conference* on Dependable Systems and Networks	[6]
V4: IET Intl. *Conference* on System Safety	[4]
V5: Intl. *Conference* on Computer Safety, Reliability, and Security	[15]
V6: *Journal* of Reliable Intelligent Environments	[2]
V7: Pacific Rim Intl. *Symposium* on Dependable Computing	[12]
V8: SEAS DTC Technical *Conference*	[1]
V9: Intl. Requirements Engineering *Conference*	[3]
V10: PeerJ Computer Science (*journal*)	[9]
V11: Intl. *Conference* on Software Engineering Advances	[7]
V12: Intl. Conference on Agile Software Development	[11]
V13: Agile *Conference*	[5]
V14: Computer and Information Science (*book series*)	[10]

Table 4. Roles involved in safety case development.

Study	Role	Construction	Assessment
Graydon et al. [6]	Developer	Build	Review
Hall and Rapanotti [7]	Customer		Approve
	Developer	Build	
	Regulator		Approve
Alexander et al. [1]	Developer	Build	Review
Ge et al. [5]	Safety engineer	Build	Review
Matsuno et al. [12]	Customer		Approve
	Developer	Build	
	Development manager		Review
	Test manager		Review
Knight et al. [8]	Developer	Build	
	System stakeholder		Approve
	Certifier/regulator		Approve
Lin and Shen [10]	Developer	Build	
Kuramitsu [9]	Owner	Build	Review
	Developer	Build	Review
	Operator	Build	Review
	User		Review
Stålhane and Myklebust [16]	Safety engineer		Review
	Customer		Review
	Developer	Build	Review
Cicotti [2]	Assurance engineer	Build	
O'Halloran et al. [13]	Developer	Build	Review
	Customer		Review
	Safety engineer		Review
Cleland-Huang and Vierhauser [3]	Develop team	Build	Review
	Safety master	Overview	Review

5 RQ5 – WHAT ARE THE CHALLENGES AND NEEDS ADDRESSED BY THE APPROACHES?

The motivation of this question is to identify the practical problems addressed by the approaches that promote continuous safety assurance alongside system development processes.

We identified many challenges and needs in the studies selected. Some studies addressed more that one challenge or need. Table 5 shows the challenges and the studies that addressed them.

Table 5. Challenges addressed

Challenge	#	%
(C1) Integrate safety assurance practices into the SDLC	7	43.7%
(C2) Dealing with system complexity and uncertainty	4	25%
(C3) Reduce development risks and costs	3	18.7%
(C4) Reuse safety arguments/cases/documents	3	18.7%
(C5) Achieve balance between functionality and safety	2	12.5%
(C6) Safety case construction	2	12.5%
(C7) Reaching agreement among stakeholders	2	12.5%

C1 – Integrate safety assurance practices into the SDLC: The most noted challenge was how to find ways of integrating safety assurance practices into the SDLC. Ruiz et al. [15] indicated that it is still unclear how to handle safety concerns during development phases. Dick and Wills [4] pointed out that the gathering of evidence needs to occur throughout the SDLC. Cicotti [2] observed that current practices address the risk management processes separately from the SDLC, thus leaving vendors to define and manage the interactions between such processes. Stålhane and Myklebust [16] and Łukasiewicz and Górski [11] noted that including SAC construction into an agile process is a challenging and costly change. Cleland-Huang and Vierhauser [3] added as challenging and costly the incremental development of new features and certification of the modified system. O'Halloran et al. [13] indicated that many standards already require the integration of safety engineering activities within the SDLC in an early, iterative and continuous basis.

C2 – Dealing with system complexity and uncertainty: The second most noted challenge acknowledges that systems complexity or uncertainty may impose great risks in case of SAC production separated from the SDLC. Alexander et al. [1] focused on autonomous systems and noted how difficult it is to assure the safety of unpredictable behaviour systems. It may be unfeasible to made a major up-front design, and also it may only be possible to assess system based on emergent behaviour and the rationale behind the system architecture. Such context calls for integrated engineering of safety assurance and software. Ruiz et al. [15] and Cicotti [2] noted that the increasing complexity of systems affects the safety integrity of solutions and make it difficult to analyse the multiple environmental conditions properly. Dick and Wills [4] considered as demanding the coordination and organisation of the collection, review, and publication of certification evidence.

C3 – Reduce development risks and costs: Some studies were concerned with better management of risks and costs throughout the system and safety case development. Hall and Rapanotti [7] and Graydon et al. [6] pointed out that development errors that weaken the safety assurance argumentation may be discovered early in the process, provide that the argumentation development occurs alongside the software. Finding errors as early as possible reduce risks and late rework efforts. Lin and Shen [10] reported that the postponement of SAC construction is time-consuming and error-prone.

C4 – Reuse safety arguments/cases/documents: Three studies indicated that incremental SAC could foster reuse of SACs artefacts. Ge et al. [5] focus on reuse of safety arguments across releases, suggesting an emergent specification approach. Stålhane and Myklebust [16] indicates that existing SAC holds quality and safety management argumentation that would not change substantially on following releases or products. O'Halloran et al. [13] noticed the difficulties in incorporating third-party components and their corresponding safety artefacts into the SCS project, highlighting the need to carefully manage assumptions and expectations across supply chains.

C5 – Achieve a balance between functionality and safety: Two studies addressed the impracticality of taking apart functionality and safety. Graydon et al. [6] noted that satisfying functional requirements goals is not sufficient in the context of safety-critical development. On the other side, a system that is safe but fails to satisfy requirements goals is also unacceptable. Knight et al. [8] stated that assuring that all engineering elements in the life cycle contribute to the assurance of system safety. In a value-based perspective, implemented functions without proper assurance argumentation are of marginal value.

C6 – Safety case construction: Easing the composition of safety arguments and evidence was cited by two studies. Papadopoulos and McDermid [14] highlighted the significant variation that SAC go through across industries and the potential for generic approaches for certifications. Matsuno et al. [12] pointed out that writing cases, decomposing arguments and choosing evidence are still challenging.

C7 – Reaching agreement among stakeholders: How to better use SACs to reach agreement among stakeholders was cited by two studies. Matsuno et al. [12] indicated that more active participation of stakeholders could improve the incremental review of SACs. Kuramitsu [9] noted that the involvement of non-experts stakeholders in arguments review could be better investigated.

6 RQ6 – WHICH ARE THE TECHNIQUES AND TOOLS USED FOR SAFETY CASE DEVELOPMENT?

We aim to identify the structuring and assessment techniques used by existing approaches and to analyse if the way of structuring and assessment of SACs impacts its continuous development. We also want to identify the level of tool support that those approaches provide.

Safety case is a category of Argumentation-Induced evidence structuring technique [20]. There are many ways to construct and assess them. In this section, we overview the notation languages, assessment techniques, and tools used in the studies.

Table 6 depicts the notation languages adopted by the selected studies (each consider only one notation). Most works adopt GSN notation, which is indeed very popular in literature [20]. TCL, POE, D-Case and a textual notation proposed by [18] were cited once each. TCL language is a CAE-like notation supported by NOR-STA tool. D-Case is based on GSN with some extensions to have a broader use for dependability cases. POE is a textual notation used to organize problem statements and its decomposition into sub-statements and evidence of solutions. Holloway's notation is one of the five textual templates proposed as alternatives to the graphical notations [18]. Three studies did not discuss a specific notation [14][1][8].

Table 7 shows the classification of the assessment techniques according to the categories proposed by Nair et al. [20]. Most studies explicitly indicated the review of argumentation as a Qualitative assessment technique (10 studies - 67%). Four studies did not explicitly discuss any assessment technique. Although we can consider that at some point there will be a human-based review of the safety case, it is due that proposals address when, how and who would perform such reviews.

Two studies discussed assessment techniques categorized as Checklist. Graydon et al. [6] proposed a process framework that prompts developers to review their candidate development choices using a set of criteria. This checking drives the developer to establish confidence in the argument or to

Table 6. Structuring techniques.

Notation	Notation type	Studies
GSN	Visual	[6][5][10][15] [9][2][13] [3]
DOORS/TraceLine	Visual	[4]
POE	Textual	[7]
D-Case	Visual	[12]
Holloway's notation	Textual	[16]
TCL	Visual	[11]
Not discussed		[14][1][8]

improve it. Knight et al. [8] introduced an analysis framework that treats a system and its safety case as a single entity. The framework comprises a list of statements about the safety properties of the system; as the development progress, the statements serve as a checklist.

Only one study discussed a Logic-based technique. Knight et al. [8] provided a notification mechanism that performs in-production monitoring of safety properties of the system. The study also proposes an audit phase (Qualitative) in its comprehensive life cycle, thus including three different assessment techniques. Matsuno et al. [12] discussed dependability metrics (Quantitative) managed throughout the dependability cases development, with support of the AssureNote tool. The study also proposed a qualitative review of the argumentation to complement the quantitative assessment. The advancement towards more non-qualitative techniques will foster a more systematic way of incremental development.

Table 7. Assessment techniques (RQ5).

Assessment technique	Assessment category	Studies
Argument review	Qualitative	[14][4][7][12] [8][10][9][16][2][13][3]
Development choice criteria	Checklist	[6]
Analysis framework	Checklist	[8]
Monitoring mechanism	Logic-based	[8]
Dependability metrics	Quantitative	[12]
Not discussed		[1][5][15][11]

Three tools appeared in the studies. Matsuno et al. [12] used the D-Case Editor[1], a visual tool that supports GSN and D-Case descriptions. Łukasiewicz and Górski [11] used the NOR-STA[2] tool that is a management system for supporting conformance with norms and standards; it supports safety case documentation in CAE notation. Kuramitsu [9] used the AssureNote[3], a GSN authoring tool. Other 12 studies did not disclose the tool support used (if any). Lack of tool support is still noticed as a gap between theory and practice [21], although there were found 46 assurance case tools in a recent survey [19]. There is a need to provide more empirical data and discussion towards tools for assurance cases.

SMS REFERENCES

[1] R. Alexander, T. Kelly, and B. Gorry. 2010. Safety Lifecycle Activities for Autonomous Systems Development. In *4th SEAS DTC Technical Conf.* Edinburgh, 1323–1330.

[1] https://github.com/d-case/d-case_editor

[2] http://www.nor-sta.eu/en

[3] https://github.com/AssureNote/AssureNote

[2] G. Cicotti. 2017. An evidence-based risk-oriented V-model methodology to develop ambient intelligent medical software. *Journal of Reliable Intelligent Environments* 3, 1 (2017), 41–53.

[3] J. Cleland-Huang and M. Vierhauser. 2018. Discovering , Analyzing , and Managing Safety Stories in Agile Projects. *IEEE 26th Intl. Requirements Engineering Conf.* (2018), 262–273.

[4] A. Dick and S. Wills. 2008. Evidence-Based Development - Applying Safety Engineering Techniques to the Progressive Assurance and Certification of Complex Systems. In *2008 3rd IET Intl. Conf. on System Safety*. 1–6.

[5] X. Ge, R. Paige, and J. McDermid. 2010. An Iterative Approach for Development of Safety-Critical Software and Safety Arguments. In *Agile Conf.* 35–43.

[6] P. Graydon, J. Knight, and E. Strunk. 2007. Assurance Based Development of Critical Systems. In *37th Annual IEEE/IFIP Intl. Conf. on Dependable Systems and Networks (DSN'07)*. IEEE, 347–357.

[7] J. Hall and L. Rapanotti. 2008. Assurance-Driven Design. In *The Third Intl. Conf. on Software Engineering Advances*. 379–388.

[8] J. Knight, J. Rowanhill, M. Aiello, and K. Wasson. 2015. A Comprehensive Safety Lifecycle. In *Intl. Ws. on Assurance Cases for Software -Intensive Systems*, Floor Koornneef and Coen van Gulijk (Eds.). Springer, 38–49.

[9] K. Kuramitsu. 2016. Continuously revised assurance cases with stakeholders ' cross-validation : a DEOS experience. *PeerJ Computer Science* 2 (2016), e101.

[10] C. Lin and W. Shen. 2015. Generation of assurance cases for medical devices. *Studies in Computational Intelligence* 566 (2015), 127–140.

[11] K. Łukasiewicz and J. Górski. 2018. Introducing Agile Practices into Development Processes of Safety Critical Software. In *Proc. of the 19th Intl. Conf. on Agile Software Development: Companion (XP '18)*. ACM, 1–8.

[12] Y. Matsuno, J. Nakazawa, M. Takeyama, M. Sugaya, and Y. Ishikawa. 2010. Towards a Language for Communication among Stakeholders. In *2010 IEEE 16th Pacific Rim Intl. Symposium on Dependable Computing*. 93–100.

[13] M. O'Halloran, J. Hall, and L. Rapanotti. 2017. Safety engineering with COTS components. *Reliability Engineering and System Safety* 160 (2017), 54–66.

[14] Y. Papadopoulos and J. McDermid. 1999. The potential for a generic approach to certification of safety critical systems in the transportation sector. *Reliability Engineering and System Safety* 63, 1 (1999), 47–66.

[15] A. Ruiz, P. Barbosa, Y. Medeiros, and H. Espinoza. 2015. Safety Case Driven Development for Medical Devices. In *Intl. Conf. on Computer Safety, Reliability, and Security (SAFECOMP)*, Vol. 9337. Cham, 183–196.

[16] T. Stålhane and T. Myklebust. 2016. The agile safety case. *Lecture Notes in Computer Science* 9923 LNCS (2016), 5–16.

REFERENCES

[17] D. Dermeval, J. Vilela, I. Bittencourt, J. Castro, S. Isotani, P. Brito, and A. Silva. 2016. Applications of ontologies in requirements engineering: a systematic review of the literature. *Requirements Engineering* 21, 4 (01 Nov 2016), 405–437.

[18] C. Holloway. 2008. Safety Case Notations: Alternatives for the Non-Graphically Inclined?. In *2008 3rd IET Intl. Conf. on System Safety*. 1–6.

[19] M. Maksimov, N. Fung, S. Kokaly, and M. Chechik. 2018. Two Decades of Assurance Case Tools: A Survey. In *Intl. Workshop on Assurance Cases for Software-Intensive Systems (ASSURE)*, Barbara Gallina, Amund Skavhaug, Erwin Schoitsch, and Friedemann Bitsch (Eds.). Springer, 49–59.

[20] S. Nair, J.L. De La Vara, M. Sabetzadeh, and L. Briand. 2014. An extended systematic literature review on provision of evidence for safety certification. *Inf. and Software Technology* 56, 7 (2014), 689–717.

[21] S. Nair, J.L. De La Vara, M. Sabetzadeh, and D. Falessi. 2015. Evidence management for compliance of critical systems with safety standards: A survey on the state of practice. *Inf. and Software Technology* 60 (2015), 1–15.

[22] S. Tiwari and A. Gupta. 2015. A systematic literature review of use case specifications research. *Information and Software Technology* 67 (2015), 128 – 158.